

The Model answer

درجہ اولیہ = اول کلاس

Question 1: Choose the correct answer:

(15 marks)

- 1) Allows objects of different classes related by inheritance to respond differently to the same message:
a) Polymorphism b) Inheritance c) Multiple Inheritance d) Single Inheritance
- 2) The class which contains at least one pure virtual function are known as:
a) Base Class b) Parent Class c) Pure Class d) Abstract Base Class
- 3) Which of the following true as a pointer variable:
a) int *intptr; b) int * intptr; c) int* intptr; d) All are true.
- 4) Go to end of existing file; write anywhere:
a) ios::app b) ios::ate c) ios::in d) ios::out
- 5) A pointer that contains address is called:
a) A Null pointer b) A Zero pointer c) Delete pointer d) Error
- 6) The syntax of operator overloading is:
a) type operator sign (parameters) { /* body */ } b) type sign operator (parameters) { /* body */ }
c) type operator + sign (parameters) { /* body */ } d) None of these
- 7) Default open mode of file created from an ifstream class is:
a) Open file for read b) Open file for read and write
c) Open file for write d) Has no default.
- 8) This is a constructor for an object:
a) Rectangle Rect(12, 20); b) Rectangle Rect={ 12, 20};
c) Rectangle Rect; d) All are true.
- 9) In the public inheritance:
a) The private in parent is in accessible by child b) The private in parent is accessible by child
c) The protected in parent is in accessible by child d) The public in parent is in accessible by child.
- 10) Access Specifiers are:
a) Can be listed in any order in a class b) If not specified, the default is private
c) Can appear multiple times in a class d) All are true.

11) In this statement: `me.open("myfile.txt", ios::out|ios::in);`

- a) File is an object from fstream class
- b) File is an object from ofstream class
- c) File is an object from ifstream class
- d) File is not created

12) Is a standard notation for the modelling of real-world objects as a first developing an object oriented program:

- a) A class
- b) An Object
- c) Unified Modelling Language
- d) Data Abstraction

13) Which of the following is true for a constructor:

- a) Has no return type
- b) Must have return value
- c) Can not be overloaded
- d) Only one per class

14) Can be accessed by the base class (parent) objects and the derived class (child) objects too:

- a) Multiple inheritance
- b) Polymorphism
- c) Hidden methods
- d) None of these is true.

15) Statement allocates an array of characters:

- a) `char *pstr = new char[sizeof(str)];`
- b) `char *pc = new char('a');`
- c) `char *pstr = new char[sizeof('a')];`
- d) None of these is true.

Question 2: State which of the following are true and which are false: (15 marks)

- 1) When using nested structure you must define the inside structure first (T).
- 2) Dynamic memory is allocated using operator new (T).
- 3) A class is made abstract by declaring that class virtual (F).
- 4) Creating a structure allocates a memory space for each element in the structure (F).
- 5) Different variables of a fundamental type (int) are applied the addition operator (T).
- 6) Private access specifier can only be called by or accessed by functions that are members of the class (T).
- 7) Union is not similar to a struct, and all members share a single memory location in both of them (F).
- 8) We can compare two struct variables directly like this : `if (s1 >= s2)` (F).
- 9) Virtual function and pure virtual function must have a definition in the class (F).
- 10) The user must write an initialization list for the structure (F).
- 11) All virtual functions in an abstract base class must be declared as pure virtual functions (F).
- 12) Different variables of a Point class are applied the addition operator (F).
- 13) Comparing addresses in pointers is not the same as comparing contents pointed at by pointers (T).
- 14) Each variable in program is stored at a unique address (T).
- 15) Single inheritance is a derived class which inherits from multiple base class (T).

11) in this statement: `me.open("myfile.txt", ios::out|ios::in);`

- a) File is an object from fstream class
- b) File is an object from ofstream class
- c) File is an object from ifstream class
- d) File is not created

12) Is a standard notation for the modelling of real-world objects as a first step in developing an object oriented program:

- a) A class
- b) An Object
- c) Unified Modelling Language
- d) Data Abstraction

13) Which of the following is true for a constructor:

- a) Has no return type
- b) Must have return value
- c) Can not be overloaded
- d) Only one per class

14) Can be accessed by the base class (parent) objects and the derived class (child) object, too:

- a) Multiple inheritance
- b) Polymorphism
- c) Hidden methods
- d) None of these is true.

15) Statement allocates an array of characters:

- a) `char *pstr = new char[sizeof(str)];`
- b) `char *pc = new char('a');`
- c) `char *pstr = new char[sizeof('a')];`
- d) None of these is true.

Question 2: State which of the following are true and which are false: (15 marks)

- 1) When using nested structure you must define the inside structure first (T).
- 2) Dynamic memory is allocated using operator new (T).
- 3) A class is made abstract by declaring that class virtual (F).
- 4) Creating a structure allocates a memory space for each element in the structure (F).
- 5) Different variables of a fundamental type (int) are applied the addition operator (T).
- 6) Private access specifier can only be called by or accessed by functions that are members of the class (T).
- 7) Union is not similar to a struct, and all members share a single memory location in both of them (F).
- 8) We can compare two struct variables directly like this : `if(s1 >= s2)` (F).
- 9) Virtual function and pure virtual function must have a definition in the class (F).
- 10) The user must write an initialization list for the structure (F).
- 11) All virtual functions in an abstract base class must be declared as pure virtual functions (F).
- 12) Different variables of a Point class are applied the addition operator (F).
- 13) Comparing addresses in pointers is not the same as comparing contents pointed at by pointers (T).
- 14) Each variable in program is stored at a unique address (T).
- 15) Single inheritance is a derived class which inherits from multiple base class (T).

Question 3:

(15 marks)

1. What is the difference between the inside (**inline**) function and the outside function declaration in the class definition. Show the memory size for each. Explain your answer with example?

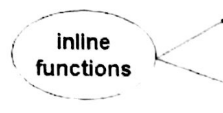
The Answer:

- Member functions defined inside the class declaration are called inline functions
- Only very short functions, like the one below, should be inline functions

```
int getSide()  
{ return side; }
```

Inline function

```
class Square  
{  
    private:  
        int side;  
    public:  
        void setSide(int s)  
        { side = s; }  
        int getSide()  
        { return side; }  
};
```



Defining Member Functions After the Class Declaration

- Put a function prototype in the class declaration
- In the function definition, precede function name with class name and scope resolution operator (::)

```
int Square::getSide()  
{  
    return side;  
}
```

- Inside the Class Declaration:

- ▶ Member functions defined inside the class declaration are called inline functions
- ▶ Only very short functions, like the one below, should be inline functions

```
int getside ( ){ return side; }
```

- Outside the Class Declaration:

- ▶ Member functions defined outside the class declaration used the scope operator (binary operator)

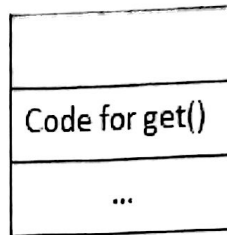
```
Method_type  class Name :: Method name ()  
{ definition }
```

```
int Square :: getside()  
{ return side; }
```

- Difference between inside (inline) and outside class implementation:

```
(1) Class A {
    int x;
    public:
    int get() {return x;}
};
```

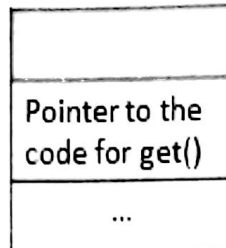
Memory for object of A



x
get() Size=size of (x)+size of (get)
Memory size ↑ speed ↑

```
(2) Class B {
    int x;
    public:
    int get() ;
};
int B :: get()
{return x;}
```

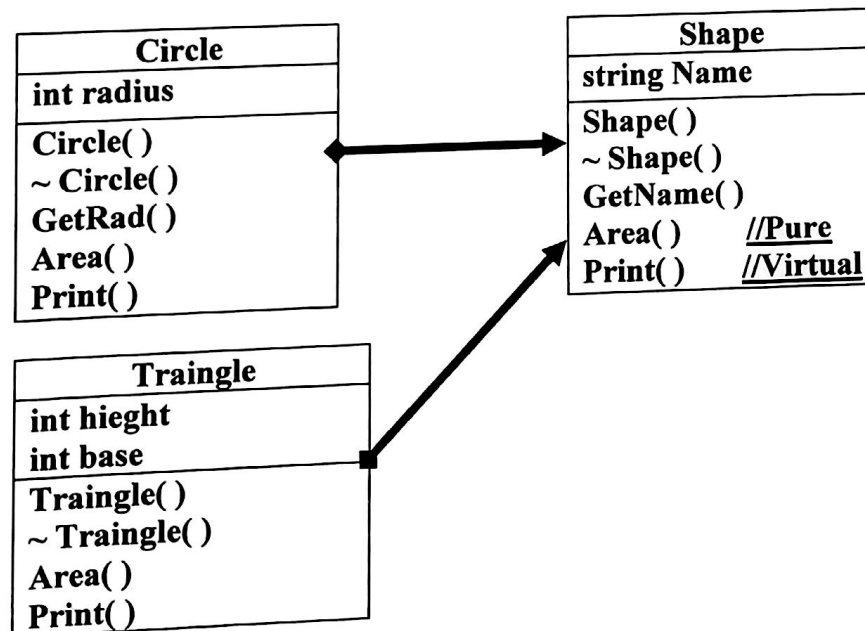
Memory for object of B



x
Size=size of (x)+size of (Pointer)
Memory size ↓ speed ↓

If size(inline code)=size (Pointer)Then inline implementation very good and faster
If size(inline code)>size (Pointer)Then inline implementation bad

2. Write a complete C++ program to implement and fully declare the classes represented by the following UML:



The Answer:

```
// virtual members
#include <iostream>
using namespace std;
class Shape {
protected:
    int width, height;
public:
    void set_values (int a, int b)
        { width=a; height=b; }
    virtual int area()
        { return 0; }
};
class Rectangle: public Shape
{
public:
    int area()
        { return width*height; }
};
class Triangle: public Shape
{
public:
    int area()
        { return width*height/2; }
};
```

```
int main ()
{
    Rectangle rect;
    Triangle trgl;
    Shape poly;
    Shape * ppoly1 = &rect;
    Shape * ppoly2 = &trgl;
    Shape * ppoly3 = &poly;
    ppoly1->set_values (4,5);
    ppoly2->set_values (4,5);
    ppoly3->set_values (4,5);
    cout << ppoly1->area() << '\n';
    cout << ppoly2->area() << '\n';
    cout << ppoly3->area() << '\n';
    return 0;
}
```

```
// Abstract Class
class Shape {
protected:
    int width, height;
public:
    void set_values (int a, int b) { width=a; height=b; }
    virtual int area()=0;
};
```

Question 4:

(15 marks)

1. Complete the implementation of the following String class. Note that two versions of the constructor and `=` are required, one for initializing/assigning to a String using a `char*`, and one for string initialization/assignment. **Operator + should concatenate two strings:**

```
class String{
private:
    char *ch;
    int len;
public:
    String (char*);
    String (String &);
    ~String ( );
    String& operator= (char*);
    String& operator= (String&);
    int Length( )
        {return (len);}
    String operator +(String, String);
};
```

The Answer:

```
#include "stdafx.h"
#include <iostream>
#include <memory>
using namespace std;
class String{
private:
    char * chars;
    int len;

public:
    String()
    {
        len=0;
    }

    String(char * s)
    {
        chars=s;
        strlen(s);
    }

    String & operator = (char* s)
    {
        chars=s;
        len=strlen(s);
        return *this;
    }
```

```
String & operator +(String s)
{
    char * temp= new char[strlen(chars)+strlen(s.chars)+1];
    strcpy(temp,chars);
    strcat(temp,s.chars);
    chars=temp;
    len=strlen(temp);
    return *this;
}

int Length()
{
    return len;
}

};
```

2. Rewrite the following C++ program after correcting the errors (it has 5 errors). Write the output after updating the errors:

1	#include <iostream>	19	Point Point ::operator - (Point p)
2	using namespace std;	20	{
3	class Point:	21	Point temp;
4	{	22	temp.x = x - p.x;
5	{	23	temp.y = y - p.y;
6	public:	24	return temp;
7	int x,y;	25	}
8	Point () {x=0; y=0;};	26	int main ()
9	Point (int a,int b) { x=a; y=b; }	27	{
10	Point operator + (Point);	28	Point P1 (3,6);
11	Point operator - (Point);	29	Point P2;
12	};	30	Point sum, sub;
13	Point Point ::operator+ (Point p);	31	sum = P1 + P2;
14	{	32	sub = P1 - P2;
15	Point temp;	33	cout << sum.x << ',' << sum.y << '\n';
16	temp.x = x + p.x;	34	cout << sub.x << ',' << sub.y << '\n';
17	temp.y = y + p.y;	35	return 0;
18	return temp;	36	}
	}		

With my best wishes,
Dr. Dina M. Ibrahim